Pastel

Network Working Group
Request for Comments #269
NIC # 7817
Categories: File Transfer
Updates: 122, 238, 172
Obsoletes: None

H. Brodie
UCLA-NMC
6 December 71

# Some Experience with File Transfer

At UCLA-NMC we have recently completed implementation of a program which utilizes UCSB's storage capability via their Simple Minded File System (See RFC #122 by Jim White for a description of SMFS). The use of the program is detailed in Appendix A.

We learned a number of things in the implementation effort and subsequent usage. We think a number of these apply towards the development of a net-word-wide File Transfer Protocol, and we hope to stimulate further discussion of these issues. We discovered some things in the UCSB protocol that we would like to see included in the network-wide protocol, and we see some things that are in the currently proposed net protocol and are unfortunately absent from the UCSB protocol.

In the first category, is the UCSB file retrieval procedure. The user specifies among other things, a bit count of the number of bits to be retrieved in the current request.

Successive RTF commands yield successive pieces of the file. Portions of the file can be spaced over by use of the SPF command. We think that the ability of the user to specify the size of the "chunks" of the file he is about to receive, along with the ability to access any part of the file without having to get the entire file, is definitely an advantage. It makes the user programs easier to write since the problem of parsing the input stream virtually disappears, as the user program knows exactly what to expect at all times. In addition, the one request, one response nature of the protocol avoids the problem of sending a request and then receiving pieces of data of unpredictable size at unknown intervals. The response to each RTF gives the comforting knowledge that "they're still listening". This leads us to believe that there is much to be gained by the adoption of a one request, one response type of protocol. We might point out that for any significant file transfer, this does not seriously cut down on the overall data transfer rate, since a "defaulting" type of approach, as SMFS uses can be used to keep the request messages very small. We also have found the mandatory password scheme of UCSB quite easily used, and any server site, (i.e. a site specifically advertising file storage) can reasonably be expected, in our opinion, to require passwords (see also RFC #238 by R. Braden).

Almost immediately after starting to use SMFS we found a serious lack in one area. There is no way for a user to ask "what files do I have there?" As a matter of fact, the author suspects that there are already several files

H. Brodie
UCLA-NMC
6 December 71

there which he has "forgotten" about! It is reasonable, perhaps even necessary, for any server to supply a nicely formatted character string describing the files stored there for this password, or user, or whatever division is used. The listing should also contain other pertinent information, such as date created, size etc.

In the meantime, UCSB is providing the SEX system with valuable "out-of-line on-line" storage, and we look forward to the development of a widely accepted file transfer protocol, implemented on a large number of server sites, hopefully equipped with low cost bulk storage.

SRX Notebook
Section 23.29

# Appendix A.

H. Brodie
3 December 71

## FXFER

FXFER is a re-entrant program written in Symbol which is used to send a receive file from UCSB, using their Simple Minded File System. For a description of the Network interface to SMFS, see RFC #122 by Jim White.

Files are stored there in a paged format identical to the paged format that the tape process uses: <header><data page><data page>... Thus a file which Master lists as having n pages will take 2048 n bytes of storage at UCSB. The user's sign-on initials are used as both the access and modification passwords, so that if a file is sent under one user's sign-on, they can only be retrieved or deleted by him.

Commands

PA - Increments "Print All" counter. Different settings of this counter yield different levels of program trace output on the console.

NP - No Print. This sets the "Print All" counter to zero.

DF FOO - This deletes the file "FOO" located at UCSB.

PF FOO - This sends the file "FOO", pointed to by the user's root directory, to UCSB. Only read access is needed to "FOO".

GF FOO - The file "FOO" located at UCSB is retrieved and put in the user's root (with write access). Note that it must have been sent by the user to have the right password.

RN FOO BAR - Rename the file "FOO" at UCSB to "BAR". The same password restrictions as with PF and GF apply. (Not yet implemented)

ST - Status. Tells what program is doing and how much its done.

X - Exit program.

If the system and/or UCSB is particularly slow, then UCSBFIL may type a time-out message. At this point the user has the option of continuing, or exiting the program. The message is self-explanatory, as are most of the program's messages.